Purpose-based Expert Finding in a Portfolio Management System

Xiaolin Niu, Gordon McCalla, Julita Vassileva[1]

Department of Computer Science
University of Saskatchewan
Saskatoon, Saskatchewan, S7N 5A9 Canada
{xin978, mccalla, jiv}@mail.usask.ca

Abstract. Computer systems are making the process of expert finding more feasible and effective than ever before, especially in peer-to-peer distributed e-business environments. This paper outlines purpose-based expert modelling as an approach for finding an expert in a multi-agent portfolio management system in which autonomous agents develop expert agent models independently and do not adhere to a common representation scheme. This approach aims to develop a taxonomy of purposes that define a variety of context-dependent user modelling processes which are used by personal agents for users to find appropriate expert agents to advise users on investing strategies.

1 Introduction

Information needs and expertise needs prompt expert seeking [17]. Expert finding has been investigated extensively in the agent community. Most of the research concerns building profiles of experts, which users can search on demand [1][6][11][16]. The effort spent on constructing and maintaining profile information of the experts is significant and expensive since they rely on often controversial information, coming from clients who may be satisfied or dissatisfied with the services of an expert because they have different preferences and criteria. Moreover, in the real world, the expert profiles might be exaggerated by the experts in order to attract more clients. We have chosen to tackle these problems. To introduce our approach, let us first look at an example scenario.

An example scenario
Suppose Bank A offers customers a 24-hour on-line portfolio management service (investment in stocks, bonds, mutual funds, etc.) where a software agent represents each customer to invest/manage his/her portfolio. In order to decide which broker agent to choose to invest/manage portfolio for his/her, the investor might want to investigate the expertise of the broker agents by asking (through their agents) other users who have had contact with some broker agents about their reputation. In this

[1] This work is supported by the Natural Sciences and Engineering Research Council of Canada.

way, customers can get more adequate information about the real capabilities of the broker agents, since each broker agent may exaggerate its advertised expertise in order to attract as many clients as possible.

■

This example highlights an important issue related to expert finding: constructing expertise models based on the distributed information kept by variety of peers in the environment. In a distributed, autonomous agent-based environment, autonomous agents keep expert model fragments, which are private users' evaluations of the expert's performance and can be used by others only if the agents are willing to share the information. These fragments cannot be expected to adhere to the same representation scheme imposed by a centralized expert model [2][5](the same problem arises in distributed databases, see [7]). Even if they are willing to entrust their private models to a centralized database, the expert model fragments come from a range of sources (e.g. raw data, other agents) and are dependent on the context (for what purpose the model was created, when and who created it), so it would be very hard to ensure consistency in a centralized expert model representation based on these fragments as input. Therefore in this kind of distributed environment, the traditional centralized expert model is replaced by expert model fragments, developed by the various software agents populating the environment for particular purposes [10], [14], [15]. Since information about each expert is distributed throughout the system, doing full integration of the information would be too expensive, and often impossible. However, it would be possible to integrate specific data, which is relevant and useful to a specific purpose, i.e. to compute an expert model at the moment and for the specific purpose it is needed. Therefore, the task of expert modelling is shifting from the collection at one place of as many data about an expert as possible to collecting on demand whatever information is available at this moment and interpreting it for a particular *purpose*. Thus, the main focus of expert modelling shifts from traditional representation issues such as consistency maintenance to issues such as determining what knowledge to retrieve for a given purpose and making sense of this knowledge in context. Expert modelling in this kind of system is a "just in time" [9] process, invoked as a part of achieving a particular purpose and using information relevant to that purpose [3].

This paper describes an approach for expert finding which is based on defining a taxonomy of expert modelling purposes. This will allow retrieving expert information relevant to a particular purpose in order to assemble and integrate fragmented expert model information. The representation of each purpose is procedural and contains a description of the context (e.g. which agents are available at the moment to provide information, what kind of user models they can provide and how much time is available for computation) in which the procedure can be applied to achieve the purpose. The purposes are retrieved and executed by distributed autonomous agents to compute user models "just in time" as they are needed. Similarly to developing a full ontology of a domain, envisaging all possible purposes for user modelling in all possible contexts is an impossible task. Therefore, the effort of the designer should focus on creating a library of important, reusable purpose clichés.

The rest of this paper reports our work in developing purposes in the portfolio management expert finding domain. We begin with a brief overview of the system architecture in section 2. Then section 3 describes the nature of a purpose, the purpose hierarchies and algorithms. Our arguments for purpose reuse are outlined in section 4. Finally, we will demonstrate several possible system architectures and conclude this paper by pointing out future research directions.

2 Multi-agent Portfolio Management System

This portfolio management system is built on a multi-agent architecture as shown in Fig. 1 based on the system in [13]. There are two kinds of agents: personal agents (PA) and expert agents (EA). Each investor has his/her own personal agent. The PA will collect an investor's risk-return preferences and other characteristics through questionnaires and game playing [13]. The PA also needs to hire the most suitable expert agent for its investor to manage assets on the investor's behalf.

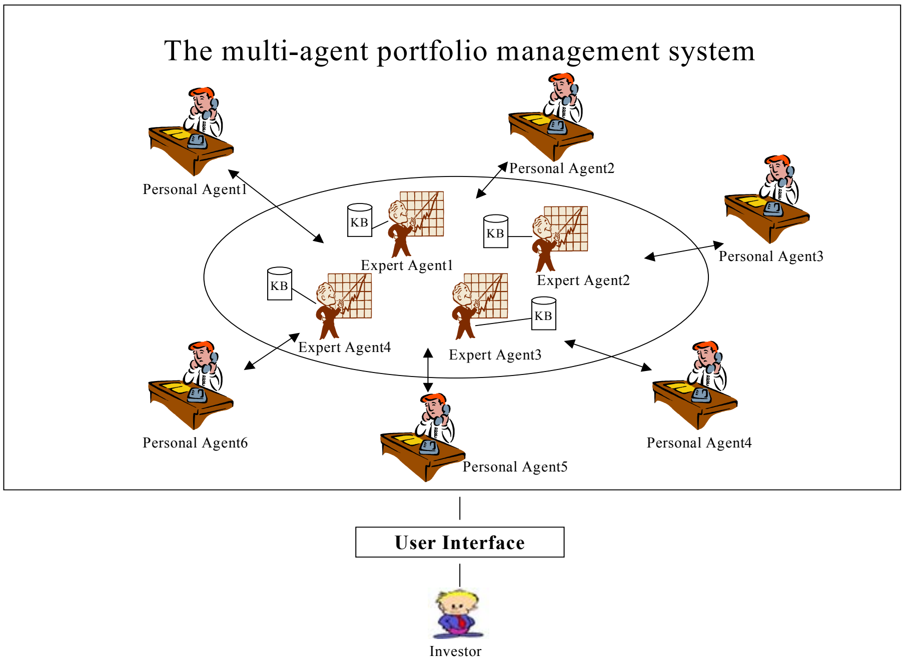

Fig.1. The architecture of the multi-agent portfolio management system

The expert agent can choose a sequence of portfolios over time to achieve a measure of performance that is appropriate to the risk-return preferences of an investor. Each EA has a knowledge base that stores different strategies for different risk-return user types. Each EA also has a risk-return preference, thus determining the kind of user appropriate for the EA. The EA can broadcast its advertisement, for example, “I am good for a risk-seeking person!” or “I am an expert for a risk-averse person!”

An example scenario is described below. An investor wants to have a diversified portfolio but lacks time for active investment. He/she believes the on-line brokers will possess superior information and uses a PA to manage his/her assets. First the PA will collect personal information about the investor through questionnaires and game playing and decide the risk return preference of the investor. Then the PA will choose the most appropriate EA for this investor based upon his/her risk-return preference, the advertisement of the EAs and the evaluation about the EAs from other PAs who have been hired by the investors who are friends of the investor or are with similar risk-return preferences. After a period of investment, an evaluation form needs to be filled out by the investor to evaluate the performance of the EA. The result will be stored in the PA for further evaluation needs. From this example, therefore, the user/agent models, such as characteristics of an investor, risk-return preference of an investor and investor’s trust in an expert agent, are created and maintained by each autonomous personal agent, i.e. there is no single user/agent model associated with the individual expert agent. Rather there are many different “snapshots” of one expert agent taken by different agents in different contexts. The user and agent models are not only physically distributed throughout the system, but also logically decentralized since there is no standard representation scheme for these models (personal agents may be developed by different providers and are heterogeneous). Agents communicate through messages using a shared ontology and protocol, but have no access to (and even if they had, they wouldn’t be able to interpret) each other’s individual representations. It is necessary to allow agents to request from each other and integrate decentralized model fragments stored privately by each of them and to generate a desired expertise model. In next section, the purpose-based user/agent modelling approach will address this problem.

3 Purpose-based User / Agent Modelling Approach

A purpose contains three kinds of information: inputs, functions and outputs. The inputs denote the type of raw data (describing context and domain variables), which is relevant to the given purpose. The functions are algorithms used to compute the desired outputs using the inputs within context-specific resource constraints. The outputs are the result of computation and can be considered to be context-specific partial user/agent models. These partial models can also form input to other purposes.

An important feature of purposes is that they can be organized into *hierarchies*. The library of purposes can be viewed at many levels, for example, from very general to very specific. There are two ways to organize purposes: *generalization* and *aggregation* (similar to plans produced using hierarchical planning [4]).

In a generalization hierarchy (e.g. as in Fig. 2), the specific purposes inherit information and procedures from more general purposes in the hierarchy. One specific purpose in the stock investment domain is *Purpose-1*, which is to select an appropriate expert agent for an investor. A higher-level purpose is to select an agent (not necessarily an expert agent) to match the needs of a person (not necessarily an investor). Two more specific purposes are to find an expert agent for a retired investor and to find an expert agent for a student investor.

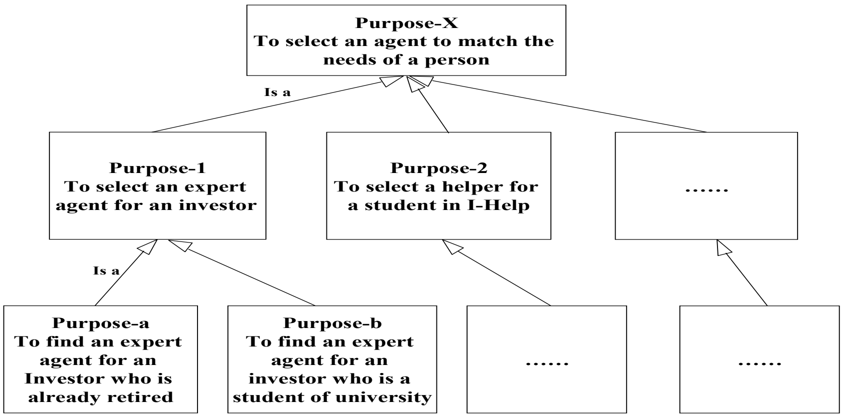

Fig. 2. An example of generalization hierarchy

Purposes can also aggregate sub-purposes, resulting in aggregation hierarchies (e.g. Fig. 3) where some sub-purposes can be part of several super purposes. The way sub-purposes are aggregated can be defined by the functions of the super purpose. For example, *Purpose-1* matches the model of the investor and the EA by consecutively integrating user/agent model fragments computed as results of 4 sub-purposes (as shown in Fig.3). The four sub-purposes of *Purpose-1* can be called in one by one depending on resource and time availability. The algorithm in *Purpose-1* can be stopped at any time and will generate the best answer so far (depending on which of the sub-purposes were executed). However, the sub-purposes of *Purpose1-1* (and the other second level purposes) do not have this anytime aspect, since they require fully completing each sub-purpose.

Consider one of the sub-purposes of *Purpose1-2*, ***Purpose1-2c***: to calculate the rating R^2 of each expert agent. This purpose calculates the rating of each expert agent based on what proportion of a given group of personal agents uses the expert agent. Let's denote with $\beta_E \in [0,1]$ the evidence of usage. For example, if EA_1 (Expert Agent 1) is used by 75% of the personal agents, then β_1 is 0.75. Therefore, the rating R^2 for each expert agent so far can be derived using the rating computed by *Purpose1-1* (which computes the rating of each expert agent based on the difference between the advertisement of the expert agent and user type of the investor), i.e. $R^1{}_E$ and the new evidence β_E. The simple reinforcement learning formula is:

$$R^2_E = \varepsilon\, R^1_E + (1 - \varepsilon)\, \beta_E$$

where $0 \le \varepsilon \le 1$ is a coefficient which denotes how much the agent values the new evidence β_E. The output from *Purpose1-2c* is a vector $[R^2_1 \ldots R^2_n]$, where R^2_i denotes the rating of the i^{th} expert agent.

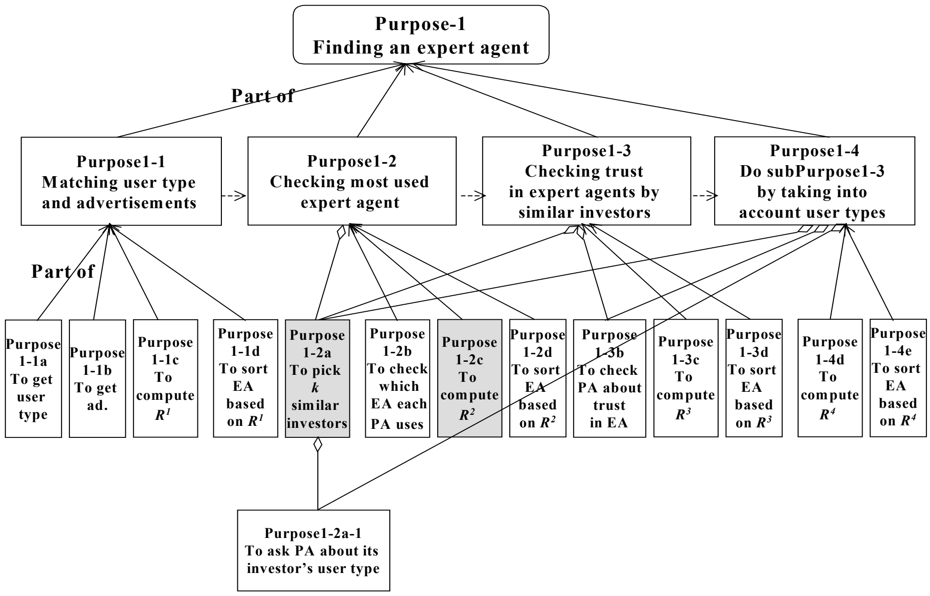

Fig. 3. The aggregation hierarchy for *Purpose-1*, finding an expert agent

4 Purpose Re-use

Purposes defined in this way form a repository of clichés, which can be reused to expand the range of purposes for the application or for different applications. There are several ways to reuse the purposes:

- *Generalization*: A purpose can be generalized into a higher-level purpose, which can be used in different domains (as in Fig. 2).
- *Specialization*: A purpose can be specialized into a more specific purpose by specifying more constraints in additional sources of information that can be used in a specific context (as in Fig. 2).
- *Modification*: A purpose can be modified in order to adapt to a new domain where the available input data are of different types. For example, *Purpose-1* could be

adapted for the domain of peer help [8, 15] to choose a helper for a student who needs some help.

- *Sharing a purpose*: A sub-purpose can be shared in aggregation by several super-purposes. For example, *Purpose1-2a* is to pick the personal agents of investors who are of similar user type. This purpose can be re-used by three super purposes (as in Fig. 3). Once this purpose is created, designer effort is saved when other super purposes re-use this sub-purpose.

Purpose re-use is valuable from a software engineering point of view to save time and effort [12]. If an existing solution can be reused, this saves time that would otherwise be spent on the creation of similar or identical software components. Another motivation is flexibility in response to new requirements. Purposes can be selected by the designer and tailored to the specific needs of the application by changing some parameters, such as inputs and context information, etc.

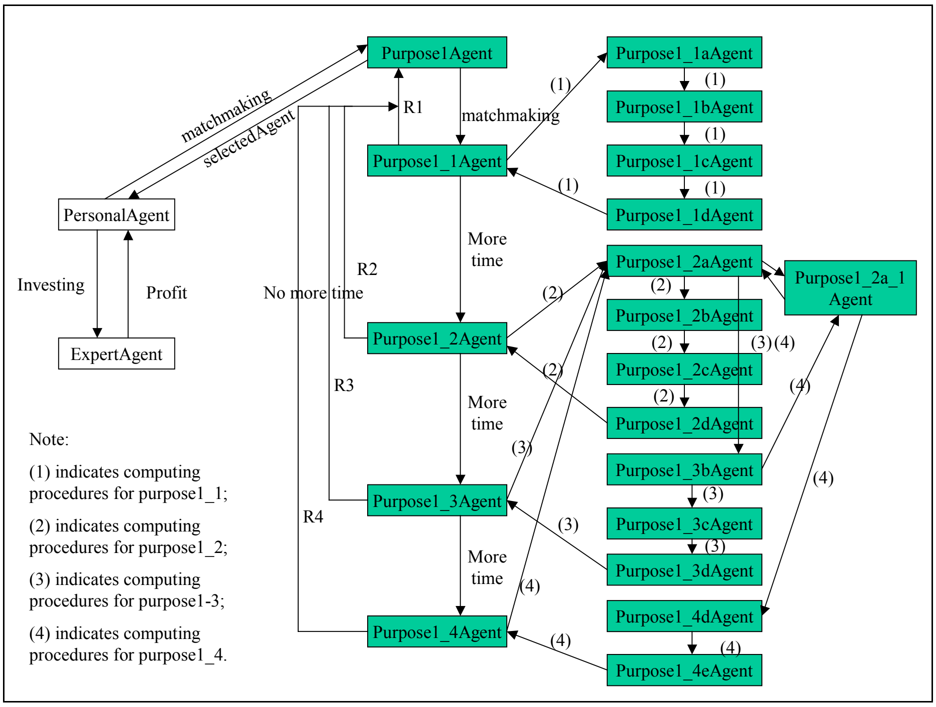

Fig. 4. The multi-agent architecture

5 Multi-agent system architecture for purpose-based user modelling

This multi-agent system is built on JADE (Java Agent Development Framework) – a software framework aimed at developing multi-agent systems and applications. The purpose hierarchies within this system architecture are maintained by a set of specialized user/agent modelling agents associated with each purpose. These agents are organized according to the purpose hierarchies. Each user/agent modelling agent knows where to find the next such agent to continue the computation needed for the next (either according to the aggregation or to the generalization dimension) sub-purpose according to the purpose hierarchy (Fig. 4). Personal or application agents subcontract user/agent modelling tasks to these specialized user/agent modelling agents, which perform computations upon request and return the results to the requesting agent without storing any data.

In this way, the computation of user models and the storage of user data in this architecture are fully distributed. Specialized purpose agents can be reused easily.

6 Conclusions and Future Work

We are currently implementing a comprehensive purpose hierarchy to support expert finding in the portfolio management system. The quality of decisions made by the agents will be evaluated with simulated users. However, demonstrating that active expert modelling helps achieve better decisions, is only a "proof of solution existence", and doesn't show the advantages of the purpose based user/agent modelling approach versus centralized matching of profiles of experts and customers. The strongest argument for the purpose-based modelling approach is that it relieves the system from the need to maintain expert profiles and provides community-based evidence to the customers. Comparing with centralized user/expert modelling, this approach implies fewer constraints on the agents (with respect to shared representation scheme, reliance on a connection to a server etc.) and is more robust (no central point of failure). It also allows for better protection of privacy, since the user models as well as their private expert models are kept locally by the personal agents and can be shared according to pre-set policies by the users. We feel that the weakest point for the active approach is the practicality of developing comprehensive reusable purpose hierarchies. Our hope is that in the future, much as ontology research is leading to comprehensive shared vocabularies for many domains, a set of overlapping expert modelling purpose cliché hierarchies will be devised for many domains and will be used to carry out active expert finding by heterogeneous software agents.

References

1. Ackerman, M. S., et al.: Recommenders for Expertise Management. In Workshop on Recommender Systems, Algorithms and Evaluation at the ACM SIGIR'99, Berkeley, CA. 1999
2. Browne, D., Totterdell, P., Norman, M.: Adaptive user interfaces. Academic Press Ltd., London, UK, 1990
3. Bull, S., Greer, J.E., McCalla, G., Kettel, L., Bowes, J. User modelling in I-Help: what, why, when and how. Proceedings of 8th International Conference, UM 2001, Sonthofen, Germany, July 2001, Pages 117-126
4. Corkill, D.: Hierarchical Planning in a Distributed Environment. In Proceedings of the Sixth International Joint Conference on Artificial Intelligence, (1979), 168-175
5. Fink, J., and Kobsa, A.: A Review and Analysis of Commercial User Modeling Servers for Personalization on the World Wide Web. User Modeling and User-Adapted Interaction, 10 (3-4) (2000), 209-249
6. Foner, L. N.: Yenta: A Multi-agent Referral-Based Matchmaking System. In the proceedings of the First International Conference on Autonomous Agents (Agents'97), Marina del Rey, CA, (1997) 301-307
7. Giunchiglia, F., Zaihrayeu, I.: Making peer databases interact - a vision for an architecture supporting data coordination. Technical Report # DIT-02-0012. Also to appear in Proc. Cooperative Information Agents (CIA 2002), Madrid, September 2002
8. Greer, J., McCalla, G., Cooke, J., Collins, J., Kumar, V., Bishop, A. and Vassileva, J.: The Intelligent HelpDesk: Supporting Peer Help in a University Course. In Proc. ITS'98, San Antonio, Texas, LNCS No1452, Springer (1998), 494-503
9. Kay, J.: A scrutable user modeling shell for user-adapted interaction. Ph.D. Thesis. Baser Department of Computer Science, University of Sydney, Sydney, Australia. (1999)
10. McCalla, G., Vassileva, J., Greer, J. and Bull, S.: Active Learner Modeling. Proc. ITS2000, Springer LNCS 1839, (2000) 53-62
11. McDonald, D. W. & Ackerman, M. S.: Expertise Recommender: a Flexible Recommendation System and Architecture. CSCW 2000, Proceeding of the ACM 2000 Conference on Computer Supported Cooperative Work, Philadelphia, PA, December 2-6, 2000, 231-240
12. Suctcliffe, A.: The Domain Theory: Patterns for Knowledge and Software Reuse. Lawrence Erlbaum Associates, Publishers Mahwah, New Jersey, ISBN: 0805839518, 2002
13. Tang, T., Winoto, P. & Niu, X.: Who Can I Trust? Investigating Trust between Users and Agents in a Multi-agent Portfolio Management System. In AAAI-2002 Workshop on Autonomy, Delegation, and Control: From Inter-agent to Groups. Edmonton, Canada, July 28, 2002
14. Vassileva, J., McCalla, G., Greer, J.: Multi-Agent Multi-User Modelling. To appear in User Modelling and User-Adapted Interaction, (2003), 28 pp manuscript
15. Vassileva, J.I., Greer, J.E., McCalla, G.I.: Openness and disclosure in multi-agent learner models. In Morales R. and Kay J. (Eds.) Proceedings of the Workshop on Open, Interactive, and Other Overt Approaches to Learner Modelling, International Conference on AI in Education, Lemans, France. (1999)
16. Vivacqua, A. S.: Agents for Expertise Location. In the Proceedings of the AAAI Spring Symposium on Intelligent Agents in Cyberspace, Stanford, CA, March 1999
17. Yimam, D. and Kobsa, A.: Expert Finding Systems for Organizations: Problem and Domain Analysis and the DEMOIR Approach. In Ackerman, M., Cohen, A., Pipek, V. and Wulf, V. eds.: Beyond Knowledge Management: Sharing Expertise. Boston, MA: MIT Press, 2000